App:

Index.tsx:

import React from "react";

import { StyleSheet, Text, View } from 'react-native'

import Stack\_Navigation from "../components/stack";

export default function Index() {

    return (

        <Stack\_Navigation></Stack\_Navigation>

    );

}

Component:

Bai1:

import { Pressable, StyleSheet, Text, View ,Image, TextInput} from 'react-native'

import React, { useState } from 'react'

import { useNavigation } from 'expo-router'

const bai1 = () => {

    const navigation = useNavigation();

    const [name,setName] = useState("");

  return (

  <View style={styles.view1}>

      <View style={styles.view2}>

          <Image style={styles.img} source={(require('.././assets/images/book.png'))}>

          </Image>

      </View>

      <Text style={styles.text}>

        MANAGE YOUR

        TASK

      </Text>

      <View style={styles.view3}>

          <TextInput placeholder='Enter your name' style={styles.ip}

            value={name}

            onChangeText={setName}

          >

          </TextInput>

          <Image style={styles.img1} source={require('.././assets/images/mail.png')}></Image>

      </View>

      <View style={styles.view4}>

            <Pressable style={styles.pre} onPress={()=>{

              navigation.navigate('bai2',{name})

            }}>

                  <Text style={styles.text1}>GET STARTED</Text>

                  <Image style={styles.img2} source={require('.././assets/images/back.png')}></Image>

            </Pressable>

      </View>

  </View>

  )

}

export default bai1

const styles = StyleSheet.create({

  view1:{

    flex:1,

   alignContent:'center',

   alignItems:'center'

  },

  view2:{

    justifyContent:"center",

    alignItems:'center',

   margin:20

  },

  img:{

    width:271,

    height:271,

    resizeMode:'contain'

  },

  text:{

    fontSize:20,

    fontWeight:'bold',

    color:'purple',

    width:150,

    textAlign:'center',

  },

  view3:{

   textAlign:'center',

   top:20

  },

  img1:{

    width:25,

    height:25,

    bottom:35,

    margin:5

  },

  ip:{

    width: 250,

        height: 30,

        borderWidth: 1,

        padding: 20,

        paddingLeft: 50,

        textAlign:'center'

  },

  view4:{

    backgroundColor:'#00BDD6',

    borderRadius:10,

    width:130,

    alignItems:'center',

    top:20,

  },

  text1:{

   fontSize:15,

   fontWeight:'400',

   top:12,

   color:'white'

  },

  img2:{

    width:25,

    height:25,

    left:90,

    bottom:10,

  }

})

Bai2:

import { StyleSheet, Text, View, Image, TextInput, FlatList, Pressable } from 'react-native'

import React from 'react'

import { useState } from 'react';

import { useEffect } from 'react';

import { useRoute } from '@react-navigation/native';

import { useNavigation } from 'expo-router';

const bai2 = () => {

  const [data, setData] = useState([]);

  const [search, setSearch] = useState('');

  const route = useRoute();

  const { name, job } = route.params || [];

  const navigation = useNavigation();

  const fetchData = (query = '') => {

    const url = query

      ? `https://6556cd42bd4bcef8b611a10d.mockapi.io/bai1?name=${query}`

      : 'https://6556cd42bd4bcef8b611a10d.mockapi.io/bai1';

    fetch(url)

      .then(response => response.json())

      .then(json => setData(json))

      .catch(error => console.error(error));

  };

  useEffect(() => {

    fetchData();

  }, []);

  useEffect(() => {

    if (job) {

      setData(prevData => [{ name: job, id: `${prevData.length + 1}` }, ...prevData]);

    }

  }, [job]);

  const handleSearch = (text) => {

    setSearch(text);

    fetchData(text);

  };

  const handleDelete = (id) => {

    fetch(`https://6556cd42bd4bcef8b611a10d.mockapi.io/bai1/${id}`, {

      method: 'DELETE',

    })

      .then(response => {

        if (response.ok) {

          setData(prevData => prevData.filter(item => item.id !== id));

        } else {

          console.log("loi")

        }

      })

      .catch(error => console.log('error', error));

  };

  const renderItem = ({ item }) => (

    <View style={styles.item}>

      <Pressable onPress={()=>handleDelete(item.id)}>

        <Text style={styles.text2}>xoa</Text>

      </Pressable>

      <Text style={styles.itemText}>{item.name}</Text>

    </View>

  );

  return (

    <View style={styles.view}>

      <View style={styles.view1}>

        <Pressable onPress={() => {

          navigation.navigate('bai1')

        }}>

          <Image style={styles.img} source={require('.././assets/images/back1.png')}></Image>

        </Pressable>

        <Image style={styles.img1} source={require('.././assets/images/user.png')}></Image>

        <View style={styles.view2}>

          <Text style={styles.text}>Hi {name}</Text>

          <Text style={styles.text1}>Have agrate day a head</Text>

        </View>

      </View>

      <View style={styles.view3}>

        <Image style={styles.img2} source={require('../assets/images/search.png')}></Image>

        <TextInput style={styles.ip} placeholder="search"

          value={search}

          onChangeText={handleSearch}

        ></TextInput>

      </View>

      <View style={styles.container}>

        <FlatList

          data={data}

          renderItem={renderItem}

          keyExtractor={item => item.id}

        />

      </View>

      <Pressable style={styles.pre} onPress={() => {

        navigation.navigate('bai3', { name })

      }}>

        <Image style={styles.img3} source={require('../assets/images/add.png')}></Image>

      </Pressable>

    </View>

  )

}

export default bai2

const styles = StyleSheet.create({

  view: {

    flex: 1,

  },

  view1: {

    flexDirection: 'row',

    alignItems: 'center',

    paddingTop: 20

  },

  img: {

    width: 30,

    height: 30

  },

  img1: {

    width: 50,

    height: 50,

    left: 120,

    borderRadius: 25,

    backgroundColor: 'purple'

  },

  view2: {

    flexDirection: 'column',

    left: 150

  },

  text: {

    fontSize: 20,

    fontWeight: 'bold'

  },

  text1: {

    fontSize: 15

  },

  view3: {

    flexDirection: 'row',

    top: 20,

    alignItems: "center"

  },

  img2: {

    width: 30,

    height: 30,

    left: 40

  },

  ip: {

    width: 350,

    height: 50,

    backgroundColor: 'white',

    borderWidth: 1,

    textAlign: 'center',

    borderRadius: 20

  },

  container: {

    flex: 1,

    paddingTop: 50,

    alignItems: 'center'

  },

  item: {

    padding: 20,

    borderBottomWidth: 1,

    borderBottomColor: '#ccc',

    flexDirection: 'row'

  },

  itemText: {

    fontSize: 18,

  },

  img3: {

    width: 50,

    height: 50

  },

  pre: {

    alignItems: 'center'

  },

  text2: {

    fontSize: 20,

    fontWeight: 'bold',

    right: 10

  }

})

Bai3:

import { useRoute } from '@react-navigation/native'

import { useNavigation } from 'expo-router';

import React, { useState } from 'react'

import { Pressable, StyleSheet, Text, View, Image, TextInput } from 'react-native'

const bai3 = () => {

    const route = useRoute();

    const { name } = route.params;

    const [job,setJob] = useState('');

    const navigation = useNavigation('');

    const handleFinish = () => {

        // Post job to Mock API

        fetch('https://6556cd42bd4bcef8b611a10d.mockapi.io/bai1', {

            method: 'POST',

            headers: {

                'Content-Type': 'application/json',

            },

            body: JSON.stringify({

                name: job, // assuming the job has a 'name' field

            }),

        })

        .then(response => {

            if (response.ok) {

                return response.json();

            } else {

                throw new Error('Network response was not ok');

            }

        })

        .then(data => {

            console.log('Success:', data);

            // Navigate to bai2 with the job and name

            navigation.navigate('bai2', { name, job });

        })

        .catch((error) => {

            console.error('Error:', error);

        });

    };

    return (

        <View style={styles.view}>

            <View style={styles.view1}>

                <Image source={require('../assets/images/user.png')} style={styles.img}></Image>

                <View>

                    <Text style={styles.text}>Hi{name}</Text>

                    <Text style={styles.text1}>Have agrate day a head</Text>

                </View>

                <Image source={require('../assets/images/back.png')} style={styles.img1}></Image>

            </View>

            <Text style={styles.text2}>ADD YOUR JOB </Text>

            <View style={styles.view2}>

                <Pressable style={styles.pre}>

                    <TextInput style={styles.ip} placeholder='nhap job vao day'

                        value={job}

                        onChangeText={setJob}

                    ></TextInput>

                </Pressable>

            </View>

            <Pressable style={styles.pre1} onPress={handleFinish}>

                <Text style={styles.text3}>Finish</Text>

            </Pressable>

        </View>

    )

}

export default bai3

const styles = StyleSheet.create({

    view: {

        flex: 1

    },

    view1: {

        flexDirection: 'row',

        margin: 20

    },

    img: {

        width: 50,

        height: 50,

        borderRadius: 20

    },

    text: {

        fontSize: 15,

        fontWeight: 'bold'

    },

    text1: {

        fontSize: 15,

        top: 5

    },

    img1: {

        width: 35,

        height: 35,

        left: 50

    },

    text2: {

        fontSize: 40,

        textAlign: 'center',

        fontWeight: '500',

        top: 50

    },

    pre: {

        flex: 0,

        top: 100,

        justifyContent: 'center'

    },

    ip: {

        width: 200,

        height: 40,

        textAlign:'center',

        backgroundColor:'white',

        borderWidth:2,

        borderRadius:10

    },

    view2:{

        alignItems:'center'

    },

    pre1:{

        top:200,

        alignItems:'center'

    },

    text3:{

        fontSize:30,

        fontWeight:'bold',

        backgroundColor:'red',

        width:90,

        height:50,

        borderRadius:10

    }

})

Stack:

import React from "react";

import { NavigationContainer } from "@react-navigation/native";

import { createNativeStackNavigator } from "@react-navigation/native-stack";

import bai1 from "../components/bai1";

import bai2 from "../components/bai2";

import bai3 from "../components/bai3"

const Stack = createNativeStackNavigator();

export default function Stack\_Navigation() {

    return (

        // Make sure NavigationContainer is only in the root file

            <Stack.Navigator screenOptions={{headerShown : false} }>

                <Stack.Screen name="bai1" component={bai1} />

                <Stack.Screen name="bai2" component={bai2} />

                <Stack.Screen name="bai3" component={bai3} />

            </Stack.Navigator>

    );

}

Color.ts:

const tintColorLight = '#0a7ea4';

const tintColorDark = '#fff';

export const Colors = {

  light: {

    text: '#11181C',

    background: '#fff',

    tint: tintColorLight,

    icon: '#687076',

    tabIconDefault: '#687076',

    tabIconSelected: tintColorLight,

  },

  dark: {

    text: '#ECEDEE',

    background: '#151718',

    tint: tintColorDark,

    icon: '#9BA1A6',

    tabIconDefault: '#9BA1A6',

    tabIconSelected: tintColorDark,

  },

};

Hooks:

export { useColorScheme } from 'react-native';

//

export function useColorScheme() {

  return 'light';

}

//

import { useColorScheme } from 'react-native';

import { Colors } from '@/constants/Colors';

export function useThemeColor(

  props: { light?: string; dark?: string },

  colorName: keyof typeof Colors.light & keyof typeof Colors.dark

) {

  const theme = useColorScheme() ?? 'light';

  const colorFromProps = props[theme];

  if (colorFromProps) {

    return colorFromProps;

  } else {

    return Colors[theme][colorName];

  }

}

Scripts:

#!/usr/bin/env node

/\*\*

 \* This script is used to reset the project to a blank state.

 \* It moves the /app directory to /app-example and creates a new /app directory with an index.tsx and \_layout.tsx file.

 \* You can remove the `reset-project` script from package.json and safely delete this file after running it.

 \*/

const fs = require('fs');

const path = require('path');

const root = process.cwd();

const oldDirPath = path.join(root, 'app');

const newDirPath = path.join(root, 'app-example');

const newAppDirPath = path.join(root, 'app');

const indexContent = `import { Text, View } from "react-native";

export default function Index() {

  return (

    <View

      style={{

        flex: 1,

        justifyContent: "center",

        alignItems: "center",

      }}

    >

      <Text>Edit app/index.tsx to edit this screen.</Text>

    </View>

  );

}

`;

const layoutContent = `import { Stack } from "expo-router";

export default function RootLayout() {

  return (

    <Stack>

      <Stack.Screen name="index" />

    </Stack>

  );

}

`;

fs.rename(oldDirPath, newDirPath, (error) => {

  if (error) {

    return console.error(`Error renaming directory: ${error}`);

  }

  console.log('/app moved to /app-example.');

  fs.mkdir(newAppDirPath, { recursive: true }, (error) => {

    if (error) {

      return console.error(`Error creating new app directory: ${error}`);

    }

    console.log('New /app directory created.');

    const indexPath = path.join(newAppDirPath, 'index.tsx');

    fs.writeFile(indexPath, indexContent, (error) => {

      if (error) {

        return console.error(`Error creating index.tsx: ${error}`);

      }

      console.log('app/index.tsx created.');

      const layoutPath = path.join(newAppDirPath, '\_layout.tsx');

      fs.writeFile(layoutPath, layoutContent, (error) => {

        if (error) {

          return console.error(`Error creating \_layout.tsx: ${error}`);

        }

        console.log('app/\_layout.tsx created.');

      });

    });

  });

});